

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

Listing of claims:

Claim 1 (Currently amended): A computerized method for creating an instrumented executable file, the method comprising:

- modifying an executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling fault simulation code to control execution of the original function;
- storing the address and the name ~~retaining access information~~ of the original function in a function lookup table;
- mapping the name of the executable file to the address of the function lookup table;
- storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;
- retrieving the address of the function lookup table and the address of the executable file from the master lookup table;
- locating the function lookup table in the executable file using the address of the executable file;
- retrieving the address of the original function ~~the access information~~ from the function lookup table using the name of the original function; and
- invoking the original function using the address of the original function ~~access information~~.

Claim 2 (Cancelled).

Claim 3 (Previously presented): The computerized method for creating an instrumented executable file as in claim 1, wherein the user-supplied function is in a dynamic link library.

Claim 4 (Original): The computerized method for creating an instrumented executable file as in claim 1, wherein the user-supplied function is not exported during compilation.

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

Claim 5 (Original): The computerized method for creating an instrumented executable file as in claim 1, wherein the original function and the user-supplied function have identical prototypes.

Claim 6 (Original): The computerized method for creating an instrumented executable file as in claim 1, where the user-supplied function is stored in a module that is separate from the executable file.

Claim 7 (Previously presented): The computerized method for creating an instrumented executable file as in claim 1, wherein modifying the executable file is performed using user-specified set points.

Claim 8 (Original): The computerized method for creating an instrumented executable file as in claim 7, wherein modifying the executable file further comprises determining whether the original function implements the thiscall calling convention, and when the determination is positive, adding instructions to the executable file to perform:

pushing the register that holds the 'this' pointer onto the stack from the invoked original function site when the determining indicates that the function implements a thiscall calling convention; and

swapping the return value of the invoking original function on the stack and the register that holds the 'this' pointer value on the stack when the determining indicates that the function implements a thiscall calling convention.

Claim 9 (Previously presented): The computerized method for creating an instrumented executable file as in claim 7, wherein modifying the executable file further comprises enabling the user-supplied function to invoke the original function in the executable file.

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

Claim 10 (Original): The computerized method for creating an instrumented executable file as in claim 9, wherein enabling the user-supplied function to invoke the original function in the executable file further comprises:

adding a jump in the user-supplied function to a function that retrieves the address of the original function; and

adding a jump in the user-supplied-function that invokes the original function using the address of the original function.

Claim 11 (Original): The computerized method for creating an instrumented executable file as in claim 1, further comprising enabling the user-supplied function to alter behavior.

Claim 12 (Original): The computerized method for creating an instrumented executable file as in claim 11, wherein enabling the user-supplied function to alter behavior is performed in response to data.

Claim 13 (Original): The computerized method for creating an instrumented executable file as in claim 12, wherein the data is retrieved from an initialization file.

Claim 14 (Previously presented): The computerized method for creating an instrumented executable file as in claim 1, wherein the retaining further comprises:

saving the address of the original function in a threaded local storage variable;

and

creating an entry in the function lookup table associating the address of the original function with the name of the original function, wherein the function lookup table is in the instrumented executable file.

Claim 15 (Currently amended): A computerized method for executing an instrumented executable file comprising:

modifying the instrumented executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling fault simulation code to control

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

execution of the original function, the user-supplied function having a jump to the original function;

saving the address and the name of the original function in a threaded local storage variable;

mapping the name of the instrumented executable file to the address of a function lookup table;

storing the name of the instrumented executable file, the address of the instrumented executable file and the address of the function lookup table in a master lookup table;

retrieving the address of the function lookup table and the address of the instrumented executable file from the master lookup table;

locating the function lookup table in the instrumented executable file using the address of the instrumented executable file;

retrieving the address of the original function from the threaded local storage variable using the name of the original function; and

invoking the user-supplied function using the address of the original function.

Claim 16 (Currently amended): The computerized method for executing an instrumented executable file as in claim 15, further comprising creating the a master lookup table at initialization wherein the master lookup table associates the base address of the instrumented executable file to the address of a function lookup table in the instrumented executable file.

Claim 17 (Original): The computerized method for executing an instrumented executable file as in claim 15:

wherein original function is in a dynamic link library; and

wherein the saving and the invoking is performed by a stub function of the original function, the stub function being located in the instrumented executable file.

Claim 18 (Original): The computerized method for executing an instrumented executable file as in claim 15:

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

wherein original function is embedded in the instrumented executable file; and
wherein the saving and the invoking is performed by the original function.

Claim 19 (Original): The computerized method for executing an instrumented executable file as in claim 15, further comprising invoking the original function from within the user-supplied function using the threaded local storage variable.

Claim 20 (Original): The computerized method for executing an instrumented executable file as in claim 19, wherein invoking the original function further comprises:
pushing the register that holds the 'this' pointer onto the stack from the invoked original function site when the determining indicates that the function implements a thiscall calling convention; and
swapping the return value of the invoking original function on the stack and the register that holds the 'this' pointer value on the stack when the determining indicates that the function implements a thiscall calling convention.

Claim 21 (Currently amended): A computerized method for instrumenting an imported function in an executable file for testing by callers of the imported function, the method comprising:

adding a wrapper of the imported function to an import data block;
adding a stub function for the imported function wherein the stub function comprises an instruction that saves the address of the imported function to a threaded local storage variable and replaces an access to the imported function with an access to a user-supplied function, the user-supplied function enabling fault simulation code to control execution of the stub function;

adding an entry in a function lookup table associated with of the imported function, wherein the entry comprises the address and the name of the imported function;
mapping the name of the executable file to the address of the function lookup
table;

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;

retrieving the address of the function lookup table and the address of the executable file from the master lookup table;

locating the function lookup table in the executable file using the address of the executable file;

retrieving the address of the imported function from the threaded local storage variable using the name of the imported function; and

invoking the imported function using the address of the imported function.

Claim 22 (Previously presented): The computerized method for instrumenting an imported function in an executable file as in claim 21, the method further comprising:

determining if the prototype of the imported function is correctly specified; and

indicating an error when the determining indicates an incorrectly specified prototype of the imported function.

Claim 23 (Currently amended): A computerized method for instrumenting an embedded function in an executable file for testing by callers of the embedded function, the method comprising:

modifying the embedded function to invoke a user-supplied function in place of the embedded function using a wrapper, the user-supplied function enabling fault simulation code to control execution of the embedded function;

adding an entry in a function lookup table, wherein the entry comprises of the address and the name of the embedded function;

mapping the name of the embedded function to the address of the function lookup table;

storing the name of the embedded function, the address of the embedded function and the address of the function lookup table in a master lookup table;

retrieving the address of the function lookup table and the address of the embedded function from the master lookup table;

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

locating the function lookup table in the executable file using the address of the embedded function:

retrieving the entry from the function lookup table using the name of the embedded function; and

invoking the embedded function using the entry.

Claim 24 (Previously presented): A computerized method for instrumenting an embedded function in an executable file as in claim 23, wherein the modified embedded function comprises an instruction that causes a jump to the user-supplied function.

Claim 25 (Original): A computerized method for instrumenting an embedded function in an executable file as in claim 23, the method further comprising:

determining whether the prototype of the embedded function is correctly specified; and

indicating an error when the determining whether the prototype of the embedded function is correctly specified indicates an incorrectly specified prototype of the embedded function.

Claim 26 (Original): A computerized method for instrumenting an embedded function in an executable file as in claim 23, wherein the function lookup table is in the executable file.

Claim 27 (Cancelled).

Claim 28 (Currently amended): A computerized system comprising:

means for modifying an executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling fault simulation code to control execution of the original function; and

means for storing the address and the name ~~retaining access information~~ of the original function in a function lookup table;

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

means for mapping the name of the executable file to the address of the function lookup table;

means for storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;

means for retrieving the address of the function lookup table and the address of the executable file from the master lookup table;

means for locating the function lookup table in the executable file using the address of the executable file;

means for retrieving the address of the original function the access information from the function lookup table using the name of the original function; and

means for invoking the original function using the address of the original function access information.

Claim 29 (Currently amended): A computerized system comprising:

an executable file having a call to an original function, the original function having an identity comprising a name and a parameter prototype;

means for modifying the executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling fault simulation code to control execution of the original function;

means for storing the address and the name of the original function in a function lookup table;

means for mapping the name of the executable file to the address of the function lookup table;

means for storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;

means for retrieving the address of the function lookup table and the address of the executable file from the master lookup table;

means for locating the function lookup table in the executable file using the address of the executable file;

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

means for configuring the user-supplied function to retrieve the address of stored
~~access information~~ of the original function using the name of the original function;

means for invoking the original function using the address of the original function
~~access information~~.

Claim 30 (Currently amended): A computerized system comprising:

an executable file having a jump to an original function, the original function
having an identity comprising a name and a parameter prototype;

a first software component having a user-supplied function that includes a jump to
the original function; and

a second software component for:

receiving the identity of the original function;

receiving the identity of the user-supplied function;

instrumenting the executable file by modifying the executable file to invoke the
identity of the user-supplied function in place of the identity of the original function, the
identity of the user-supplied function enabling fault simulation code to control execution
of the original function;

storing address and the name of the original function in a function lookup table
address in the executable file ~~in association with the name of the original instrumented~~
function;

mapping the name of the executable file to the address of the function lookup
table;

storing the name of the executable file, the address of the executable file and the
address of the function lookup table in a master lookup table;

retrieving the address of the function lookup table and the address of the
executable file from the master lookup table;

locating the function lookup table in the executable file using the address of the
executable file;

retrieving the address of the original function from the function lookup table
address using the name of the original ~~instrumented~~ function; and

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

invoking the original function using the address of the original function-address.

Claim 31 (Currently amended): A computerized system comprising:

a first module of machine-readable code comprising:

a call to an original function, the call being directed to a user-supplied function; and

a first data structure associating the identity of the original function with the location of the original function, wherein the first module is configured to:

store the first data structure in a function lookup table,

map the identity of the first module to the location of the function lookup table,

store the identity of the first module, the location of the executable file and the location of the function lookup table in a master lookup table; and

a second module of machine-readable code comprising the user-supplied function linked to the first module and a jump to the original function, the user-supplied function enabling fault simulation code to control execution of the original function, wherein the second module is configured to:

retrieve the location of the function lookup table and the location of the first module from the master lookup table,

locate the function lookup table in the first module using the location of the first module,

retrieve the location of the original function from the function look up table is ~~retrieved~~ using the identity of the original function, and ~~wherein~~

invoke the original function ~~is invoked~~ using the location of the original function.

Claim 32 (Currently amended): The computerized system as in claim 31,

wherein the first data structure comprises the a function lookup table, wherein the function lookup table is arranged to verify for-verifying that a threaded local storage variable contains the correct address for the original function; and

wherein the second module comprises a dynamic linked library.

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

Claims 33-35 (Cancelled).

Claim 36 (Currently amended): A computer-readable medium having computer-executable instructions to cause a computer to perform a method comprising:

- modifying an executable file to invoke a user-supplied function in place of an original function, the user-supplied function enabling fault simulation code to control execution of the original function;
- storing the address and the name retaining access information of the original function in a function lookup table;
- mapping the name of the executable file to the address of the function lookup table;
- storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;
- retrieving the address of the function lookup table and the address of the executable file from the master lookup table;
- locating the function lookup table in the executable file using the address of the executable file;
- retrieving the address of the original function ~~the access information~~ from the function lookup table using the name of the original function; and
- invoking the original function using the address of the original function ~~access information~~.

Claims 37-40 (Cancelled).

Claim 41 (Currently amended): A computer-implemented method for configuring an executable file, the executable file having an access to an original function, the computer-implemented method comprising:

- replacing the access to the original function with an access to a user-supplied function, the user-supplied function enabling fault simulation code to control execution of the original function; and

App. No. 09/503,215
Amendment dated March 22, 2005
Reply to final Office action of December 22, 2004

~~storing the address and the name retaining access information associated with the original function in a function lookup table, the user-supplied function enabling fault simulation code to control execution of the original function;~~

~~mapping the name of the executable file to the address of the function lookup table;~~

~~storing the name of the executable file, the address of the executable file and the address of the function lookup table in a master lookup table;~~

~~retrieving the address of the function lookup table and the address of the executable file from the master lookup table;~~

~~locating the function lookup table in the executable file using the address of the executable file;~~

~~retrieving the address of the original function the access information from the function lookup table using the name of the original function; and~~

~~invoking the original function using the address of the original function access information.~~

Claim 42 (Currently amended): The computer-implemented method of Claim 41, further comprising configuring the user-supplied function to invoke the original function using the address of access information associated with the original function.

Claim 43 (Previously presented): The computer-implemented method of Claim 41, wherein replacing the access to the original function with the access to the user-supplied function is performed by modifying the executable file.

Claim 44 (Previously presented): The computer-implemented method of Claim 41, wherein replacing the access to the original function with the access to the user-supplied function is performed by modifying set points stored in a computer-readable medium separate from the executable file.

Claims 45-47 (Cancelled).